To appear in *IEEE Transactions on Pattern Analysis and Machine Intelligence*

Adjusting Shape Parameters using Model-Based Optical Flow Residuals

Douglas DeCarlo[*]
and
Dimitris Metaxas[†]

Abstract

We present a method for estimating the shape of a deformable model using the least-squares residuals from a model-based optical flow computation. This method is built on top of an estimation framework using optical flow and image features, where optical flow affects only the motion parameters of the model. Using the results of this computation, our new method adjusts all of the parameters so that the residuals from the flow computation are minimized. We present face tracking experiments that demonstrate that this method obtains a better estimate of shape compared to related frameworks.

Index terms: non-rigid shape and motion estimation, model-based optical flow, deformable models

1 Introduction

Applications of model-based face tracking require both identifying users and interpreting their actions. Only by watching faces can conversational interfaces [5], interactive kiosks [24] and robots [11] start to understand more features of natural face-to-face dialogue. And only by gathering accurate motion estimates of faces can facial animation systems be automated [27] or videos of faces be compressed effectively [15, 16, 27]. Even though some of these systems do not need to know about the user's appearance, having an accurate estimate of the face shape is still important,

[*] D. DeCarlo is with the Department of Computer Science and Center for Cognitive Science, Rutgers University, New Brunswick, NJ. E-mail: decarlo@cs.rutgers.edu

[†] D. Metaxas is with the Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA. E-mail: dnm@central.cis.upenn.edu

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE **Adjusting Shape Parameters using Model-Based Optical Flow Residuals**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Rutgers University,Department of Computer Science,New Brunswick,NJ,08903**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE		**26**	
unclassified	**unclassified**	**unclassified**			

as it makes the system more accurate, efficient and robust. This argument can be made for most domains which lend themselves to model-based vision.

A model of a viewed object invites one to distinguish parameters which describe the object's underlying and unchanging shape from those which describe its motion—temporary non-rigid deformations away from this shape. Making such a categorization is a first step towards the difficult problem of simultaneously estimating shape and motion. Because the interpretation of shape and motion from an image is highly ambiguous, it is also important to consider how parameters are informed by the observations.

Consider a model-based optical flow computation, for example. Observed motion can directly constrain only the motion parameters; the true values of the shape parameters do not change over time. However, inaccurate shape estimates can make it impossible for the model to explain the observed motion using the motion parameters—this produces errors in the perceived motion. In this case, we can adjust the shape parameters to improve the entire estimate.

The challenge is to be faithful to the distinction between shape and motion parameters as the adjustment is computed. For example, it is insufficient to simply stage the computation, and use the leftovers from the motion estimate to feed a computation which determines how the shape parameters could have changed over time to explain the remaining observed motion [15]. This method simply treats shape parameters as motion parameters. In this paper, we propose computing an adjustment to the shape parameters that minimizes the error in the motion estimate. In other words, we determine a new configuration for which the motion parameters would have produced less error in the first place.

Using flow to estimate shape is the subject of the large body of work on structure from motion using an optical flow field, reviewed in [1]. There has also been a great deal of work on the structure from motion problem using feature correspondences, which is surveyed in [13]. Applying these techniques to tracking and estimating the shape of faces is rather difficult, as most methods are not suited to non-rigid motion; there have only been successful implementations of this quite recently

[9, 14].

This paper describes an alternative to structure from motion methods; our method is coupled to a model-based optical flow computation using a deformable model. Instead of performing direct surface reconstruction from optical flow, our method indirectly adapts model parameters so that the error in the model-based optical flow is reduced.

1.1 Separation of shape and motion

The starting point for our method is an intuitive distinction between shape and motion; to account for this distinction, the process of model design must encode information about a class of objects by categorizing parameters as describing variation in shape or motion. The shape parameters are *static* quantities for a particular observed object, and describe its unchanging geometric features. The motion parameters are *dynamic* quantities, which change when the observed object moves or deforms. Of course, there is no guarantee that the shape and motion of some class of objects is separable in an arbitrary parameterization; this is a simplifying assumption that we make, and will only apply to a certain degree of accuracy. It works quite well for models of human faces, for instance, where shape parameters describe an individual's appearance, while motion parameters encode the location of the head, as well as facial displays and expressions. This division is often built into face models [3, 6, 16, 19, 28] to simplify model construction or estimation, and has been used to facilitate learning the variability of motions for a class of objects [25].

The ultimate goal of this separation is to produce an estimation problem with lower dimension. During estimation, the change in the shape parameters should tend to zero as the shape of the observed object is established. Once this occurs, fitting need only continue for the motion parameters. Therefore, during model design, the separation into shape and motion should encode as many of the model deformations with shape parameters as possible. This decision leads to a more efficient tracking system.

For models with separate parameters for shape and motion, certain cues such as optical flow arise due to motion in the scene, and are appropriately used only for the estimation of motion parameters (and not shape parameters). While in some cases updating all the parameters (shape and motion) based on the flow can result in smaller deviations [15], this is missing the point of separating shape and motion in the first place, and is in conflict with the view of the shape parameters as having static values.

1.2 Using residuals

A significant error in the current model estimate will interfere with the optical flow estimates of the motion, since the model and image will be misaligned. For example, if the estimate of a "nose protrusion" parameter is inaccurate, the pattern of motion that the model would predict during a head turn would be incorrect in the local region of the nose. More specifically, if the estimate of how much the nose protrudes is much less than it should be, then the model would only be able to explain a fraction of the motion information in the image region nearby the nose. As seen from this example, the interference is quite systematic, which makes it possible to adjust the current shape estimate. Here, we would aim to adjust the nose protrusion parameter in a way that reduces the motion model's error (so it better explains the image). We will perform this adjustment using the *residual* from the optical flow constraint equation.

From this residual, each pixel used in the optical flow computation supplies one piece of information which is then used to determine how the parameters can be corrected to minimize the interference. However, some pixels will not supply any useful information. And even worse, many of the pixels will include distracting information resulting from optical flow linearization, optical flow constraint violations (such as lighting changes, shadows, or specularities), motion estimation errors, and noise. As a result, we must be sure to use a sufficient number of pixels, as well as to avoid adjusting the parameters based on distracting information. We have to assume that residual contributions that result from small errors in the estimated shape are significantly larger than those caused by distracting sources. We confirm this assumption empirically in the context of face tracking in Section 5.2, and it is likely to apply in other domains where tracking with model-based optical flow is successful.

The work in this paper builds on the model-based face tracking framework described in [7]. This framework uses a model-based optical flow computation as a constraint on the motion of a deformable model, which uses features (edges) to align the model with the image. Using features prevents the accumulation of tracking error, which would have otherwise been a difficulty using flow alone. With our proposed method, changes in the image are initially attributed entirely to motion, but then the error in the reconstructed motion is used to more accurately extract the parameters of the object being tracked.

After a brief review of deformable models in Section 2, we discuss existing approaches to model-based optical flow in Section 3. Section 4 describes our method for adjusting the model parameters using residuals from a model-based optical flow computation. Section 5 presents and discusses experiments which demonstrate how our technique improves the shape estimate of a tracked face.

2 Deformable models

Deformable models [18, 23, 29] are parameterized shapes that deform due to forces according to physical laws. For vision applications, physics provides a useful analogy for treating shape estimation [18], where forces are determined from visual cues such as edges in an image. The deformations that follow produce a shape that agrees with the data.

The shape of the deformable model $\mathbf{x}$ is parameterized by a time-varying vector of values $\mathbf{q}$ and is defined over a domain Ω which can be used to identify specific points on the model; a particular point on the model is written as $\mathbf{x}(\mathbf{q};\mathbf{u})$ with $\mathbf{u} \in \Omega$, although the dependency of $\mathbf{x}$ on $\mathbf{q}$ is often omitted. The goal of shape and motion estimation is to recover the value of $\mathbf{q}$ over time from a sequence of images. For this paper, we will be using the three-dimensional parameterized face model from [7].

As stated earlier, to distinguish the processes of shape estimation and motion tracking, the parameters in $\mathbf{q}$ are rearranged and separated into $\mathbf{q}_{\mathrm{b}}$ (the basic shape of the object) and $\mathbf{q}_{\mathrm{m}}$ (rigid and non-rigid motion), so that $\mathbf{q} = (\mathbf{q}_{\mathrm{b}}^{\top}, \mathbf{q}_{\mathrm{m}}^{\top})^{\top}$. Within our face model, $\mathbf{q}_{\mathrm{b}}$ describes an individual's appearance (there are about 80 shape parameters), while $\mathbf{q}_{\mathrm{m}}$ encodes the location of their head, as well as their facial displays and expressions (there are 12 motion parameters). Figure 1 shows examples of the face model undergoing various shape deformations (showing four different individuals), motion deformations (showing brow raising and frowning, smiling, and mouth opening) and finally two examples of when several deformations are applied at once. Further detail about this model can be found in [7].

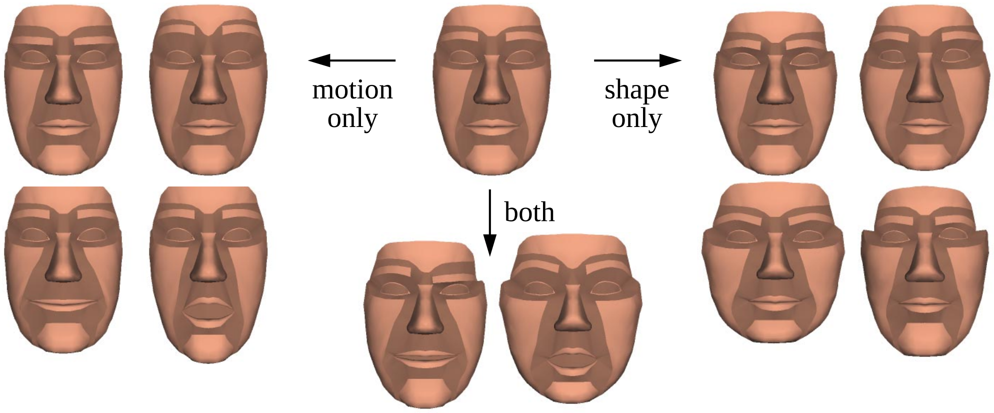


Figure 1: Example parameterized deformations of the face model (with separate parameters for shape and motion)

The model $\mathbf{x}$ is formed by applying deformation functions to the underlying shape $\mathbf{s}$. For this paper, the underlying face model $\mathbf{s}$ is a polygon mesh (shown in the center of Figure 1). There are separate deformation functions for shape ($\mathbf{T}_b$) and for motion ($\mathbf{T}_m$). The shape deformation is applied first, so that:

$$\mathbf{x}(\mathbf{q};\mathbf{u}) = \mathbf{T}_m\left(\mathbf{q}_{\mathrm{m}};\ \mathbf{T}_b\left(\mathbf{q}_{\mathrm{b}};\ \mathbf{s}(\mathbf{u})\right)\right) \tag{1}$$

The shape deformation $\mathbf{T}_b$ uses the parameters $\mathbf{q}_{\mathrm{b}}$ to deform the underlying shape $\mathbf{s}$. On top of this is the motion deformation $\mathbf{T}_m$ with parameters $\mathbf{q}_{\mathrm{m}}$, which includes a rigid translation and rotation (head motion), as well as non-rigid deformations (facial expressions and displays).

When modeling a three-dimensional object viewed in images, $\mathbf{x}$ includes a camera projection, resulting in a two-dimensional model called $\mathbf{x}_p$, which is projected flat from the original three-dimensional model.

2.1 Kinematics and dynamics

The kinematics of the model are determined in terms of the parameter velocities $\dot{\mathbf{q}}$. As the shape changes, the velocity at a point $\mathbf{u}$ on the model is given by:

$$\dot{\mathbf{x}}(\mathbf{u}) = \mathbf{L}(\mathbf{q};\mathbf{u})\dot{\mathbf{q}} \tag{2}$$

where $\mathbf{L} = \partial\mathbf{x}/\partial\mathbf{q}$ is the model Jacobian [18]. For reasons of conciseness, the dependency of $\mathbf{L}$ on $\mathbf{q}$ is often omitted.

We view $\mathbf{L}$ as consisting of components that correspond to $\mathbf{q}_\mathrm{b}$ and $\mathbf{q}_\mathrm{m}$, so that it can be written as $[\mathbf{L}_\mathrm{b}\ \mathbf{L}_\mathrm{m}]$. The Jacobian of $\mathbf{x}_p$ (the projected model) is written as $\mathbf{L}_p$, and is decomposed into components for $\mathbf{q}_\mathrm{b}$ and $\mathbf{q}_\mathrm{m}$ as $\left[\mathbf{L}_{\mathrm{b}_p}\ \mathbf{L}_{\mathrm{m}_p}\right]$. In Section 3, the image velocities associated with the optical flow are modeled by $\dot{\mathbf{x}}_{\mathrm{m}_p}$, which is the projected motion that arises due to changes in the *motion* parameters:

$$\dot{\mathbf{x}}_{\mathrm{m}_p}(\mathbf{u}) = \mathbf{L}_{\mathrm{m}_p}(\mathbf{q};\mathbf{u})\dot{\mathbf{q}}_\mathrm{m} \tag{3}$$

The shape parameters are not included, as $\dot{\mathbf{q}}_\mathrm{b}$ is not a characteristic of the scene—instead, it reflects how the shape estimate changes as more images arrive.

The models defined above are useful for applications such as shape and motion estimation when used in a physics-based framework [18]. These techniques are a form of optimization whereby the deviation between the model and the data is minimized. The optimization is performed by integrating differential equations derived from the Euler-Lagrange equations of motion. These equations are simplified in a standard manner [18], and in this case result in:

$$\dot{\mathbf{q}} = \mathbf{f}_{\mathbf{q}} \tag{4}$$

where the applied forces $\mathbf{f}_{\mathbf{q}}$ are computed from two-dimensional image forces $\mathbf{f}_{\text{image}}$ as:

$$\mathbf{f}_{\mathbf{q}} = \sum_j \mathbf{L}_p(\mathbf{u}_j)^\top \mathbf{f}_{\text{image}}(\mathbf{u}_j) \tag{5}$$

The distribution of forces on the model is based in part on forces computed from the edges of an input image [18]. With that, and given an adequate model initialization, these forces will align features on the model with image features, thereby determining appropriate parameter values. The dynamic system in (4) is solved by integrating over time, using standard (explicit) differential equation integration techniques, such as Euler integration:

$$\mathbf{q}(t+\Delta t) = \mathbf{q}(t) + \dot{\mathbf{q}}(t)\Delta t \tag{6}$$

When implementing this framework using a Kalman filter [7], (6) becomes the discrete update equation for the state. The initialization which specifies the value of $\mathbf{q}(0)$ is described in Section 5.

3 Model-based optical flow

The optical flow is typically defined as the apparent motion of brightness patterns across an image [12]. Attempting to use this information in applications such as object tracking requires assumptions about the objects (or scene) being viewed. Most common is the assumption that particular locations on viewed objects do not change in brightness. This brightness constancy assumption leads to the formulation of the well-known optical flow constraint equation at a pixel *i* in the image

I:

$$\nabla \mathrm{I}_i \begin{bmatrix} u_i \\ v_i \end{bmatrix} + \mathrm{I}_{\mathrm{t}_i} = 0 \tag{7}$$

where $\nabla \mathrm{I} = \begin{bmatrix} \mathrm{I_x} \; \mathrm{I_y} \end{bmatrix}$ are the spatial derivatives and $\mathrm{I_t}$ is the temporal derivative of the image intensity. u_i and v_i are the components of the image velocities at pixel i.

The model-based optical flow constraint equation is a reformulation of (7) in terms of a model's motion parameters $\mathbf{q}_\mathrm{m}$. When viewing a model under projection, there exists a unique model point $\mathbf{u}_i \in \Omega$ which corresponds to a particular pixel (except on occluding boundaries and situations involving transparency). In a model-based approach, the image velocities u_i and v_i are specified by projected velocities of points on the model $\dot{\mathbf{x}}_{\mathrm{m}_p}(\mathbf{u}_i)$ given by (3):

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \dot{\mathbf{x}}_{\mathrm{m}_p}(\mathbf{u}_i) = \mathbf{L}_{\mathrm{m}_p}(\mathbf{u}_i)\dot{\mathbf{q}}_\mathrm{m} \tag{8}$$

Note that only the changes resulting from the motion parameters $\mathbf{q}_\mathrm{m}$ are included, as optical flow velocities do not reflect changes in the shape parameters $\mathbf{q}_\mathrm{b}$. The model-based optical flow constraint equation is developed by rewriting (7) using (8):

$$\nabla \mathrm{I}_i \, \mathbf{L}_{\mathrm{m}_p}(\mathbf{u}_i)\dot{\mathbf{q}}_\mathrm{m} + \mathrm{I}_{\mathrm{t}_i} = 0 \tag{9}$$

When considered over a set of n pixels, a stacked set of instances of (9) can be written in matrix form as:

$$\begin{bmatrix} \nabla \mathrm{I}_1 \mathbf{L}_{\mathrm{m}_p}(\mathbf{u}_1) \\ \vdots \\ \nabla \mathrm{I}_n \mathbf{L}_{\mathrm{m}_p}(\mathbf{u}_n) \end{bmatrix} \dot{\mathbf{q}}_{\mathrm{m}} + \begin{bmatrix} \mathrm{I}_{t_1} \\ \vdots \\ \mathrm{I}_{t_n} \end{bmatrix} = \mathbf{0} \tag{10}$$

which can be written compactly as

$$\mathbf{B}\dot{\mathbf{q}}_{\mathrm{m}} + \mathbf{I}_{\mathrm{t}} = \mathbf{0} \tag{11}$$

Formulations similar to (11) (although superficially appearing quite different) can be found in [2, 15, 16, 21, 22].

An estimate of $\dot{\mathbf{q}}_{\mathrm{m}}$ (written as $\widehat{\dot{\mathbf{q}}}_{\mathrm{m}}$) using (11) is determined by solving a least-squares problem:

$$\widehat{\dot{\mathbf{q}}}_{\mathrm{m}} = \arg \min_{\dot{\mathbf{q}}_{\mathrm{m}}} \|\mathbf{B}\dot{\mathbf{q}}_{\mathrm{m}} + \mathbf{I}_{\mathrm{t}}\|^2 \tag{12}$$

Iterative approaches to solving this problem using techniques such as the Gauss-Newton method [10] are taken in [2, 15, 21, 22]. The solution in [7] performs a single step using the pseudo-inverse (where $\mathbf{B}^+$ is the pseudo-inverse of $\mathbf{B}$ [26]) [22, 16]:

$$\widehat{\dot{\mathbf{q}}}_{\mathrm{m}} = -\mathbf{B}^+\mathbf{I}_{\mathrm{t}} \tag{13}$$

This is the linear least-squares solution; it linearizes by assuming $\mathbf{L}_{\mathrm{m}_p}$ is constant (ignoring its dependency on $\mathbf{q}$). This simple solution is sufficient, as it is being combined with the result of an iterative template-alignment problem (using edges), which yields a system with higher accuracy and more robust behavior. Typical problems encountered in flow computation are avoided by strategic selection of pixels for use in (11) [7].

The derivation of (7) involves the truncation of a Taylor series, and as a result requires relatively small motions between frames. To address problems with estimating larger motions (or to implement coarse-to-fine methods), some iterative approaches transform the model geometry at each iteration using the previous motion estimate [15], while others undo the previous estimate by warping the input images [2]. The system in [7] assumes small motions (although of course could be extended using the above methods).

The most serious difficulty for these techniques, however, is combating tracking drift. Using only velocity information, small estimation errors accumulate over time. The solution to this problem is to include other information (such as features or edges) to prevent errors from building up [7, 15, 16].

3.1 Optical flow residuals

Unlike image-based optical flow techniques, these model-based methods do not require assumptions about the smoothness of the flow field to determine a solution, as the number of pixels providing useful information is sufficiently greater than the number of motion parameters. Of course, now that the solution is over-determined, there will be a residual from the least-squares solution, given the estimate $\widehat{\dot{\mathbf{q}}}_{\mathrm{m}}$:

$$\mathbf{r} = \mathbf{B}\widehat{\dot{\mathbf{q}}}_{\mathrm{m}} + \mathbf{I}_{\mathrm{t}} \tag{14}$$

The residual $\mathbf{r}$ is a vector having dimension n (the number of pixels used in the flow computation).

Contributions to the residual come from many sources. Aside from measurement noise, most obviously are linearization errors that result from ignoring the higher order terms in (7) (which were truncated in the Taylor series), and in the use of a linear least squares solution. Other contributing factors include violations of the brightness constancy assumption such as lighting changes, shadows, and specularities. Finally, shape and motion estimation errors (deviation between the current estimate of $\mathbf{q}$ and its actual value) will prevent the model from properly aligning with the image, and will cause a sizable increase in the residual.

We claim that if significant errors are present in the estimated shape and motion, they will be the primary contributors to the residual. We support this claim empirically in Section 5. This means that the residual is a valuable piece of information that can be used for estimating shape and motion, allowing us to compute small adjustments to the shape and motion parameters which reduce the residual, as we see in the next section.

4 Adjusting parameters using residuals

There are many approaches which use residuals to improve the fit of the model; we will describe three in this section. Each of these approaches involve the minimization of a distinct least-squares problem. The differences between approaches are best explained by how they respect the separation of shape and motion parameters in the model. The last approach we describe here is the main contribution of this paper—and it is also the only method that truly respects the separation between shape and motion parameters.

The naive approach treats *all* of the model parameters as motion parameters. This results in the following model-based optical flow constraint equation, as an alternative to (11):

$$\begin{bmatrix} \mathbf{B}_{\mathrm{b}} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_{\mathrm{b}} \\ \dot{\mathbf{q}}_{\mathrm{m}} \end{bmatrix} + \mathbf{I}_{\mathrm{t}} = \mathbf{B}_{\mathrm{b}}\dot{\mathbf{q}}_{\mathrm{b}} + \mathbf{B}\dot{\mathbf{q}}_{\mathrm{m}} + \mathbf{I}_{\mathrm{t}} = \mathbf{0} \tag{15}$$

where the construction of $\mathbf{B}_{\mathrm{b}}$ is analogous to $\mathbf{B}$, but uses $\mathbf{L}_{\mathrm{b}}$ instead of $\mathbf{L}_{\mathrm{m}}$. As all parameters in $\mathbf{q}$ are treated as motion parameters, one can solve for $\dot{\mathbf{q}}$ using the following:

$$\min_{\dot{\mathbf{q}}_{\mathrm{b}},\dot{\mathbf{q}}_{\mathrm{m}}} \|\mathbf{B}_{\mathrm{b}}\dot{\mathbf{q}}_{\mathrm{b}} + \mathbf{B}\dot{\mathbf{q}}_{\mathrm{m}} + \mathbf{I}_{\mathrm{t}}\|^2 \tag{16}$$

This does not respect the separation of the parameters into shape and motion as it conflicts with the static interpretation of the shape parameters. This means it can produce shape estimates that are in disagreement with the static shape of the viewed object. In practice, the treatment of all parameters as dynamic makes this method more computationally expensive and quite fragile. Further discussion on this point is made in Section 5.

Another possible approach explains the residual as directly resulting from shape deviation. In other words, the leftover motion not accounted for in $\widehat{\dot{\mathbf{q}}}_{\mathrm{m}}$ is used to update the shape with the same formulation as for determining motion (from Section 3). This is realized by the following minimization:

$$\min_{\dot{\mathbf{q}}_{\mathrm{b}}} \left\| \mathbf{B}_{\mathrm{b}}\dot{\mathbf{q}}_{\mathrm{b}} + \mathbf{B}\widehat{\dot{\mathbf{q}}}_{\mathrm{m}} + \mathbf{I}_{\mathrm{t}} \right\|^2 = \min_{\dot{\mathbf{q}}_{\mathrm{b}}} \|\mathbf{B}_{\mathrm{b}}\dot{\mathbf{q}}_{\mathrm{b}} + \mathbf{r}\|^2 \qquad (\text{given } \widehat{\dot{\mathbf{q}}}_{\mathrm{m}}) \tag{17}$$

Instead of solving one large system, the minimization in (16) is split, and is solved for motion first by (12), and then for shape in terms of the residual **r** by (17). This method is related to one described by Koch [15] where shape parameters are actually updated in two steps; first using discrepancies parallel to the line of sight, then those that are perpendicular to the line of sight.

This is a reasonable approach in the context of image-coding [15], where image fidelity is of much greater importance than the accuracy of the face shape estimate—the face shape is deformed to account for the tracking errors in motion. This produces a face shape that results in better image fidelity, but does not necessarily estimate the actual shape of the subject's face (as a result, the estimated covariance of the shape is much larger). Plus, given our distinction between shape and motion parameters, it does not make sense to adjust the shape parameters $\mathbf{q}_{\mathrm{b}}$ directly from observed velocities, since the true value of $\mathbf{q}_{\mathrm{b}}$ is a static quantity.

We take a different approach. Instead, we determine the small change in **q** that effects the largest reduction in **r**. Let $\Delta\mathbf{q}$ be the deviation between the current estimate of **q** and its true value (not including the motion extracted in $\widehat{\dot{\mathbf{q}}}_{\mathrm{m}}$). We can estimate $\Delta\mathbf{q}$ by solving the following:

$$\widehat{\Delta\mathbf{q}} = \arg\min_{\Delta\mathbf{q}} \left\| \mathbf{B}(\mathbf{q}+\Delta\mathbf{q})\,\widehat{\dot{\mathbf{q}}}_{\mathrm{m}} + \mathbf{I}_{\mathrm{t}} \right\|^2 \qquad (\text{given } \widehat{\dot{\mathbf{q}}}_{\mathrm{m}}) \tag{18}$$

How is this different? The estimate $\widehat{\Delta\mathbf{q}}$ will tell us how the shape (and motion) could have been different to produce a smaller residual in the first place. Returning to our example from the introduction, finding $\Delta\mathbf{q}$ would tell us that if the nose protruded further, the model's motion would have agreed better with the flow information.

Formally, consider what results if we assume $\Delta\mathbf{q}$ is of sufficiently small magnitude so that the first-order approximation to $\mathbf{L}_{\mathrm{m}}$ using its Taylor-series expansion is sufficiently accurate:

$$\mathbf{L}_{\mathrm{m}_p}(\mathbf{u};\mathbf{q}+\Delta\mathbf{q}) \approx \mathbf{L}_{\mathrm{m}_p}(\mathbf{u};\mathbf{q}) + \frac{\partial\mathbf{L}_{\mathrm{m}_p}(\mathbf{u};\mathbf{q})}{\partial\mathbf{q}}\Delta\mathbf{q} \tag{19}$$

We can now write down another minimization problem that is equivalent to (18) given this assumption. Combining this approximation of $\mathbf{L}_{\mathrm{m}_p}$ with the model-based optical flow constraint equation (9) results in:

$$\nabla\mathrm{I}\,\mathbf{L}_{\mathrm{m}_p}(\mathbf{u})\widehat{\dot{\mathbf{q}}}_{\mathrm{m}} + \nabla\mathrm{I}\left(\frac{\partial\mathbf{L}_{\mathrm{m}_p}(\mathbf{u})}{\partial\mathbf{q}}\Delta\mathbf{q}\right)\widehat{\dot{\mathbf{q}}}_{\mathrm{m}} + \mathrm{I}_{\mathrm{t}} = 0 \tag{20}$$

where $\partial\mathbf{L}_{\mathrm{m}_p}/\partial\mathbf{q}$ is part of the model Hessian (a rank 3 tensor). It is written here "curried" with $\Delta\mathbf{q}$ so that the parenthesized sub-expression here is a matrix.

When (20) is considered over n pixels from the input image, this results in the system:

$$\mathbf{B}\widehat{\dot{\mathbf{q}}}_{\mathrm{m}} + \left(\mathbf{G}\widehat{\dot{\mathbf{q}}}_{\mathrm{m}}\right)\Delta\mathbf{q} + \mathbf{I}_{\mathrm{t}} = \mathbf{0} \tag{21}$$

$$\text{where}\ \ \mathbf{G} = \begin{bmatrix} \left(\nabla \mathrm{I}_1 \dfrac{\partial \mathbf{L}_{\mathrm{m}_p}(\mathbf{u}_1)}{\partial \mathbf{q}}\right)^\top \\ \vdots \\ \left(\nabla \mathrm{I}_n \dfrac{\partial \mathbf{L}_{\mathrm{m}_p}(\mathbf{u}_n)}{\partial \mathbf{q}}\right)^\top \end{bmatrix} \tag{22}$$

The subscripts $[1...n]$ in the construction of $\mathbf{G}$ correspond to a particular row in (21). The transpositions performed in the construction of $\mathbf{G}$ allow it now to be curried with $\widehat{\dot{\mathbf{q}}}_{\mathrm{m}}$ (this construction transposes the second and third indices of the tensor $\mathbf{G}$). We can now rewrite (21) using the residual (14):

$$\left(\mathbf{G}\widehat{\dot{\mathbf{q}}}_{\mathrm{m}}\right)\Delta\mathbf{q} + \mathbf{r} = \mathbf{0} \tag{23}$$

which corresponds to solving the following minimization:

$$\widehat{\Delta\mathbf{q}} = \arg\min_{\Delta\mathbf{q}} \left\| (\mathbf{G}\widehat{\dot{\mathbf{q}}}_{\mathrm{m}})\Delta\mathbf{q} + \mathbf{r} \right\|^2 \tag{24}$$

Solving this least squares problem determines the best set of small changes in $\mathbf{q}_{\mathrm{b}}$ and $\mathbf{q}_{\mathrm{m}}$ that minimize the optical flow residual (14), given the linearization of $\mathbf{L}_{\mathrm{m}_p}$ in (19). In practice, we solve this using the corrected Gauss-Newton method, which performs well in cases where $(\mathbf{G}\widehat{\dot{\mathbf{q}}}_{\mathrm{m}})$ is ill-conditioned by using the singular value decomposition. In fact, the linearization in (19) is the same assumption made to justify use of the Gauss-Newton method, so we are really still solving (18). Furthermore, the use of the corrected Gauss-Newton method here makes this assumption a safe one, in terms of convergence.

The intuition for this analysis—that we're solving a minimization that is faithful to the distinction between shape and motion parameters—is realized here in the formulation of a minimization problem very different from (17). Note that it is possible that the new value of $\mathbf{r}$ would be smaller if (17) is used over (18) to estimate $\dot{\mathbf{q}}_{\mathrm{m}}$, but this goes against the assumption that $\mathbf{q}_{\mathrm{b}}$ are static parameters, and would result in an inappropriate estimate.

4.1 Updating the solution

The framework in [7] provides us with a filtered estimate of $\dot{\mathbf{q}}$ (using both the flow and edge information). The dotted region in Figure 2 shows this framework. The method from the previous section is also shown in this diagram as the block “adjustment using residuals”. We can now use the adjustment $\Delta\mathbf{q}$ by statistically combining it with the filtered solution from [7].

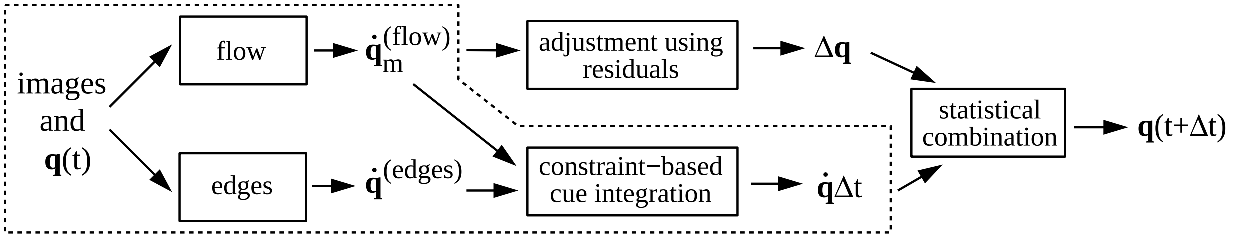

images and $\mathbf{q}(t)$
flow
edges
$\dot{\mathbf{q}}_{\mathrm{m}}^{(\mathrm{flow})}$
$\dot{\mathbf{q}}^{(\mathrm{edges})}$
adjustment using residuals
constraint−based cue integration
$\Delta\mathbf{q}$
$\dot{\mathbf{q}}\Delta t$
statistical combination
$\mathbf{q}(t+\Delta t)$

Figure 2: Schematic description of overall system

The framework in [7] uses a Kalman filter, which provides us with the covariance estimate $\Lambda_{\dot{\mathbf{q}}}$. Having uncertainty information for $\Delta\mathbf{q}$ will allow us to statistically combine these solutions, as shown towards the right in Figure 2. We model the distracting sources in $\mathbf{r}$ (aside from $\Delta\mathbf{q}$) as zero-mean Gaussian disturbances with covariance $\Lambda_{\mathbf{r}} = \sigma_{\mathbf{r}}^2\mathbf{1}$. Using (23), the (inverse) covariance of $\Delta\mathbf{q}$ is:

$$\Lambda_{\Delta\mathbf{q}}^{-1} = (\mathbf{G}\widehat{\dot{\mathbf{q}}}_{\mathrm{m}})^{\top}\Lambda_{\mathbf{r}}^{-1}(\mathbf{G}\widehat{\dot{\mathbf{q}}}_{\mathrm{m}}) \tag{25}$$

We use $\sigma_{\mathbf{r}}$ to represent the contributions to $\mathbf{r}$ from sources other than shape and motion estimation errors, and is determined in Section 5.2 from experiments where $\mathbf{q}_{\mathrm{b}}$ (the shape) is known in advance.

The statistical combination of these solutions allows the system to take into account the uncertainty in each. More specifically, it provides a principled means for the system to ignore $\Delta\mathbf{q}$ in situations when it is likely to be contaminated with background distractions. Using the covariances $\Lambda_{\Delta\mathbf{q}}$ and $\Lambda_{\dot{\mathbf{q}}\Delta t} = \Delta t^2\Lambda_{\dot{\mathbf{q}}}$, the new value of $\mathbf{q}$ is found by rewriting (23) using the typical means to combine Gaussians [4, 8], which weights $\widehat{\dot{\mathbf{q}}}\Delta t$ and $\widehat{\Delta\mathbf{q}}$ together based on their uncertainties:

$$\mathbf{q}(t+\Delta t) = \mathbf{q}(t) + \left[\Lambda^{-1}_{\dot{\mathbf{q}}\Delta t} + \Lambda^{-1}_{\Delta\mathbf{q}}\right]^{-1} \left[\Lambda^{-1}_{\dot{\mathbf{q}}\Delta t}\widehat{\dot{\mathbf{q}}}(t)\Delta t + \Lambda^{-1}_{\Delta\mathbf{q}}\widehat{\Delta\mathbf{q}}(t)\right] \tag{26}$$

This combination assumes that $\Delta\mathbf{q}$ is conditionally independent of $\dot{\mathbf{q}}$ given the images. This is a good approximation for our implementation because we use only the shape estimates in $\widehat{\Delta\mathbf{q}}$. This eliminates any overlap because $\widehat{\dot{\mathbf{q}}}_{\mathrm{m}}$ does not inform the shape estimate. Experiments showed there was little benefit in using the revised motion estimate in $\widehat{\Delta\mathbf{q}}$, as the edge solution was already quite accurate and stable.

4.2 Implementation

Solving (18) is made more efficient by omitting parameters in the construction of $\mathbf{G}$ which cannot be affected based on $\dot{\mathbf{q}}_{\mathrm{m}}$. For example, if there is no motion extracted on the forehead, then there is no reason to include eyebrow shape parameters in $\mathbf{G}$. Another example is when the extent of a parameter is simply not visible in the image. Whenever there is any motion at all, typically about half of the shape parameters of the face model can be excluded from the computations. The derivatives in $\mathbf{G}$ can be computed either analytically or numerically. We use the efficient and modular approach for analytical evaluation of these derivatives as described in [7].

The process of determining $\Delta\mathbf{q}$ can also be iterated, solving (12) and (18) repeatedly to obtain a greater improvement. For the applications here, only a single iteration is performed (the expense of further iterations isn't justified, given the small benefit they typically yield). When we do iterate this process, the algorithm does indeed converge reliably. In other applications (using a model with greater non-linearity, for instance), more stable least-squares methods could be employed (such as Levenberg-Marquardt [10]) that force convergence (although perhaps not to the global minimum).

Virtually all of the computational expense involved in using the method described in this paper results from the singular value decomposition of the matrix $\left(\mathbf{G}\widehat{\dot{\mathbf{q}}}_{\mathrm{m}}\right)$. This matrix is $n \times q$, where q is the number of parameters used in the computation; the complexity of the SVD in this case is

$O(nq^2)$.

5 Experiments

This section describes a number of face tracking experiments on two representative image sequences. We demonstrate how the adjustment method using residuals is a significant improvement over the framework in [7] and one that uses (17). We also justify our assumption that parameter estimation errors are the leading contributor to the residuals. For the remainder of this section, we will be comparing three frameworks. First, is the **original** framework from [7], which uses optical flow and edges. Second, is the **two-stage** framework based on (17), using the modification described by Koch [15] described in Section 4 which starts with discrepancies parallel to the viewing direction the follows with those that are perpendicular. Third, is the novel **residual reduction** method presented in this paper, which uses (18). The second and third methods are built using the framework of [7] as in Figure 2, where the result is statistically combined with the original solution from [7]. We were unable to compare a method based on (16), which treats all parameters as dynamic—it was too unstable (causing the system to lose track) to provide meaningful results.

The image sequences are 8 bit gray images at NTSC resolution (480 vertical lines). In the sequences, the width of the face in the image averages 200 pixels. A single subject is used in both experiments presented here. We validate the shape estimates ($\mathbf{q}_b$) using a Cyberware range scan of the subject, shown in Figure 3.

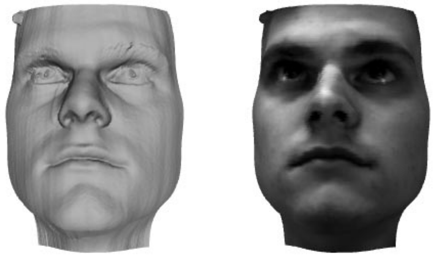


Figure 3: Range scan of the subject (shaded and textured)

The entire estimation process is automatic, except for the initialization, which requires the manual specification of several landmark features in the first frame of the sequence (the eyebrow centers, eye corners, nose tip, and mouth corners). The subject must also be at rest and (approximately) facing forward. Experience has shown that the initialization process is robust to small displacements (i.e. several pixels) in the selected landmark points. Further details of this initialization process are provided in [7].

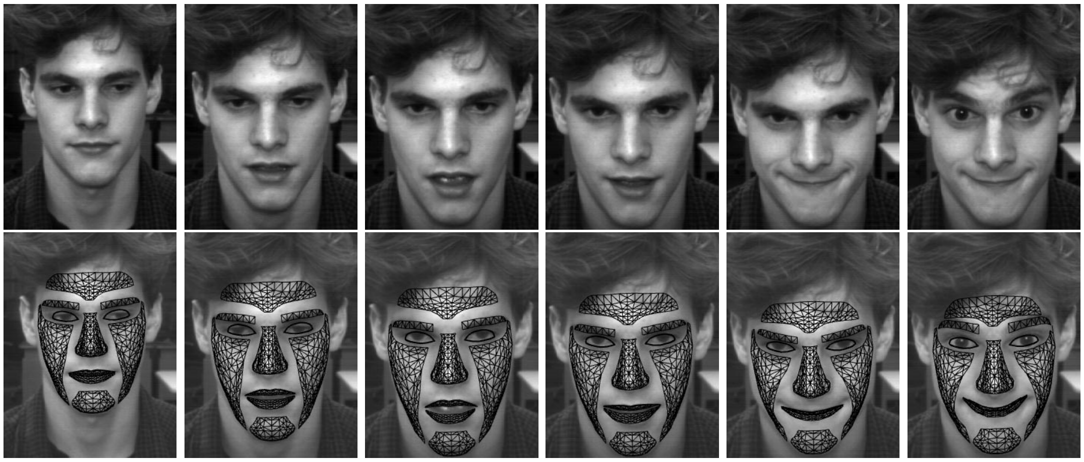
For each of the tracking examples, several frames from the image sequence are displayed, cropped appropriately. Below each, the same sequence is shown with the estimated face superimposed. Accompanying each sequence is a graph which indicates the accuracy of the shape estimate as compared to the available range scan. The graphs display the RMS error in the shape, measured at the vertices of the polygonal model. Note this includes a uniform scaling of the model so that the two faces are the same size (this eliminates the depth ambiguity—in this case, the estimated model was compared at 96% scale). As all of the approaches are initialized identically, the RMS error at the first frame is the same for all three techniques. This comparison ignores the motion parameters, so that only the shape is compared (ground truth for motion is not available).

The initialization process usually takes about 2 minutes of computation. Afterwards, processing each frame using the method in [7] takes approximately 1.4 seconds each. The two-stage approach requires an additional 3 seconds per frame, while the residual reduction method adds an additional 5 seconds per frame (all computation times are measured on a 175 MHz R10000 SGI O2). For all three methods, 120 pixels are used in the optical flow computation, selected using the methods described in [7] (using more pixels does not significantly alter the results).

5.1 Estimation experiments

The shape estimation validation experiment in Figure 4 shows the subject making a series of non-rigid face motions: opening his mouth in (b) and (c), smiling in (d) through (e), and finally raising his eyebrows in (f). At each frame, Figure 5 shows the extracted shape results as compared against the range scan of the subject, for all three techniques (the labels correspond to what was given in their descriptions above).

The RMS error starts at around 1.7 cm after initialization. For the original approach [7], the error steady declines over the course of the experiment, ending around 1.3 cm. The two-stage approach behaves similarly, but with consistently better performance than the original approach, and ending around 1.1 cm. The proposed residual reduction method performs best, ending with an RMS error of 0.85 cm. In addition, most of the adjustment took place in the first half of the sequence.



(a) Frame 1 (b) Frame 9 (c) Frame 13 (d) Frame 20 (e) Frame 24 (f) Frame 32

Figure 4: Shape estimation experiment 1

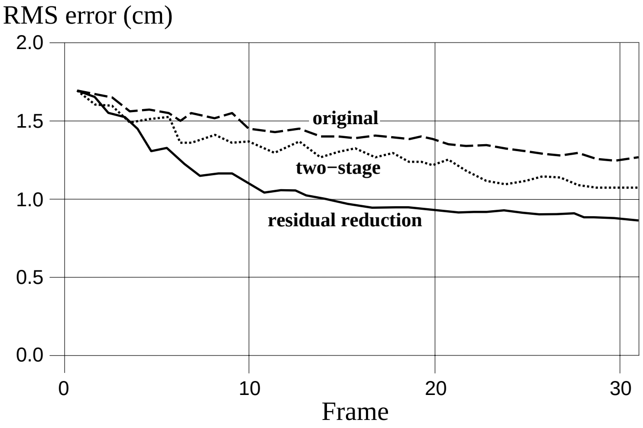
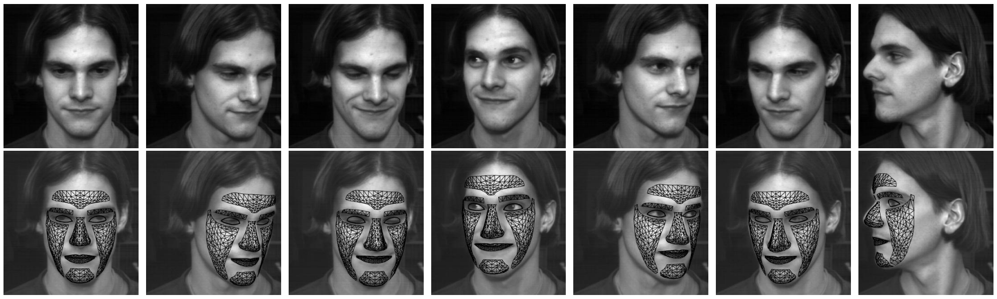


Figure 5: Shape validation of experiment 1

The experiment in Figure 6 shows the subject performing small head motions in (a) through (f) while smiling in (c) and (d), and finishing with a significant head rotation in (g). This time, the RMS error starts at around 1.9 cm after initialization. The original approach shows a gradual reduction over the sequence, ending just under 1 cm, with the large reduction in error around frame 50 corresponding to when the subject turned his head significantly to the side in Figure 6(f) and (g), where the profile view contained good edge information to fit the face shape. The two-stage approach exhibited varied performance, sometimes more and sometimes less accurate than the original solution; it ends just above 1 cm. The residual reduction method finishes with under half of the RMS error as the other techniques: around 0.4 cm. Again, this lower level was reached fairly quickly, showing another advantage of using the error residual technique.



(a) Frame 1 (b) Frame 11 (c) Frame 18 (d) Frame 24 (e) Frame 35 (f) Frame 46 (g) Frame 57

Figure 6: Shape estimation experiment 2

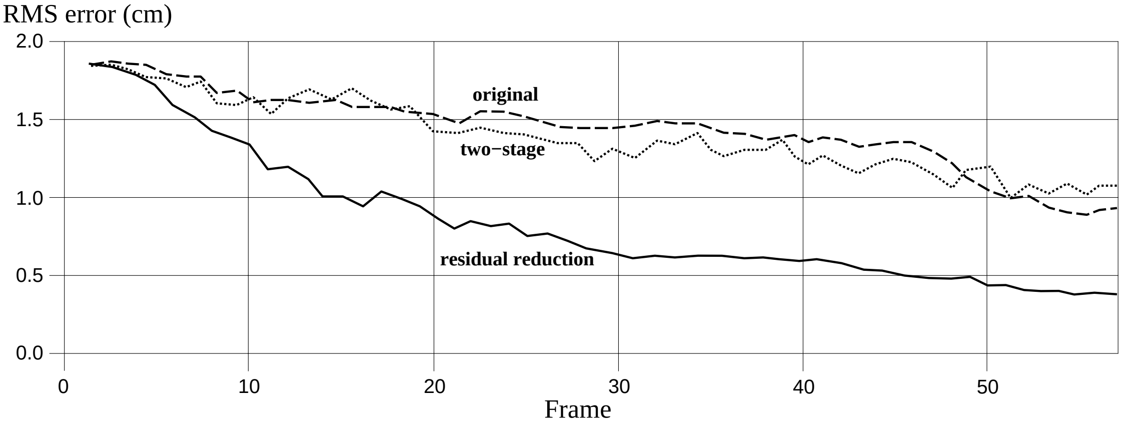


Figure 7: Shape validation of experiment 2

In all experiments, the motion parameter values change appropriately, and at the correct times. All three techniques extracted virtually the same motion parameter values. This is not particularly surprising, as the edge information will maintain fairly accurate estimates of the motion parame-

	Start	End	Start (shape given)
exp 1	0.27	0.085	0.064
exp 2	0.31	0.077	0.054

Figure 8: Residual magnitudes ($\|\mathbf{r}\|_{\text{RMS}}$) during experiments

ters. We conclude this method is primarily of use for shape estimation. However, this does not mean that the motion parameters can be omitted in the formation of **G**, as this would mean that deviations that could be best explained with motion parameters will be incorrectly explained using shape parameters.

Note that for these experiments, the background is relatively simple. This limitation originates in the use of edges from [7] for aligning the model and the image. Given the assumption of small motions by our system, the background has no effect on any of the flow computations, as the pixel selection method avoids pixels in the image that are likely to be part of the background.

5.2 Analysis of residuals

The derivation of the method using the residuals in Section 4 assumes that shape error is the leading contributor to the residuals from the motion computation. We now describe our analysis of the residuals to justify this claim by examining the residual magnitudes for the residual reduction method. First, we define the RMS residual magnitude (in pixel intensity units in the range $[0,1]$) as:

$$\|\mathbf{r}\|_{\text{RMS}} = \sqrt{\frac{1}{n}\sum_i \mathbf{r}_i^2} \tag{27}$$

Figure 8 shows the residual magnitudes at the start and end of the two experiments described earlier (in the first two columns). During both tracking experiments, the average residual magnitudes started fairly high and became considerably lower.

To isolate the portion of the residuals caused by shape error, both experiments were run again; this time, the initial model shape was taken from the range scan of the subject (shape estimation was disabled in the framework). The residuals from these two experiments turned out to be fairly constant over the entire sequence; the third column of Figure 8 contains the residual magnitudes for the first image of these experiments. By performing similar experiments on many other image sequences, we estimated the standard deviation of the distracting component of the residuals ($\sigma_{\mathbf{r}}$) to be 0.060.

We observe a dramatic reduction between the first and third columns, indicating the improved shape directly resulted in a much lower residual. It's also clear that our proposed method reduces the residual to a reasonable level (this time, compare the second and third column). This enforces the validity of our assumption that parameter estimation error is responsible for the bulk of the residual.

5.3 Discussion

Besides having improved accuracy over the original and two-stage methods, our framework extracts the shape of the face without needing data from such extreme head poses (such as a profile view). Instead, fewer observations are needed to extract the shape, so that the static part of the estimation problem converges sooner.

In addition, once the static estimation problem is complete, there is no need to perform further adjustments in our application, as there is little improvement in the motion estimates. We can also skip the adjustment computation in situations where the residual is small (compared to $\sigma_{\mathbf{r}}$), as they would tend to provide little useful information. For an application like face tracking, where the motion is the desired output (but still requires an accurate shape for good results), this is ideal.

We find that performing adjustments using this method is quite robust, at least to the same level as the underlying flow computation. In other words, the proposed algorithm almost never worsens the performance. Of course, in situations where there is large optical flow violation (such as a major lighting change), both the adjustment method and the model-based optical flow computation will fail. As shown in the validation experiments in [7], the use of multiple cues can still provide robust behavior even in some of these difficult situations.

Like in [7], we also tested our system by instigating failure in various ways. First, we added Gaussian noise directly to the images of increasing variance until the system failed (holding $\boldsymbol{\sigma}_{\mathbf{r}}$ fixed). In this case, at a noise level of 8.3% of the image intensity, the residual reduction method failed (i.e.—it produced a spurious shape estimate that later caused tracking failure). At only a slightly higher noise level (9.1%), the flow method failed. Second, we added random offsets to the initial parameters $\mathbf{q}$ (after the initialization) to see if the system could recover. This time, the flow method from the original framework failed first. We see similar behavior for sequences with large lighting changes or other violations of the optical flow constraint equation. Solutions to these problems can come on two fronts. First would be to model the situation better; for instance, using a more general form of the optical flow constraint equation to take radiometric variations into account [20]. Second, and perhaps more generally applicable, would be the use of robust techniques for cue integration, which expect some (but not all) cues or computations to fail at any time [17].

6 Conclusions

We have presented a novel deformable model technique which uses residuals from a model-based optical flow solution to refine the shape of the model. By using the relationship between the shape and motion parameterizations, small improvements to the parameters are made by minimizing the model-based optical flow residuals. It was the separation of the parameterization which made this computation possible, since additional parameters that did not apply in the model-based flow computations could still be adjusted. While this method is presented in the context of face tracking, it could certainly be applied in other model-based domains with separable parameterizations.

The adjustment computation, along with the cue integration methods from [7], seems to be fairly robust to small optical flow constraint equation violations or approximations. Besides having greater accuracy than a framework using only optical flow and edges, our framework extracts the shape of the face without needing data from extreme head poses (such as a profile view). Instead, much smaller subject motions are required to extract the shape information.

Acknowledgments

This research is supported by ARO grant DAAH-04-96-1-007; NSF Career Award (IRI-96-24604); AFOSR and ONR-YIP N00014-97-1-0817; and a RuCCS postdoctoral fellowship.

References

[1] John Barron. Computing motion and structure from noisy time-varying image velocity information. Technical Report RBCV-TR-88-24, Department of Computer Science, University of Toronto, 1988.

[2] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proceedings ECCV '92*, pages 237–252, 1992.

[3] M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *Proceedings ICCV '95*, pages 374–381, 1995.

[4] A. Bryson and Y. Ho. *Applied Optimal Control*. Halsted Press, 1975.

[5] J. Cassell. Embodied conversational interface agents. *Communications of the ACM*, 43(4):70–78, 2000.

[6] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *ECCV98*, pages II:484–498, 1998.

[7] D. DeCarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 32(2):99–127, July 2000.

[8] H.F. Durrant-Whyte. Consistent integration and propagation of disparate sensor observations. *International Journal of Robotics Research*, 6(3):3–24, 1987.

[9] P. Fua. Regularized bundle-adjustment to model heads from image sequences without calibration data. *International Journal of Computer Vision*, 38(2):153–171, July 2000.

[10] P. Gill. *Practical Optimization*. Academic Press, 1982.

[11] J. Heinzmann and A. Zelinsky. Building human-friendly robot systems. In *Proceedings of International Symposium of Robotics Research (ISRR' 99)*, 1999.

[12] B. Horn. *Robot Vision*. McGraw-Hill, 1986.

[13] T.S. Huang and A.N. Netravali. Motion and structure from feature correspondences: A review. *PIEEE*, 82(2):252–268, February 1994.

[14] T. Jebara, A. Azarbayejani, and A.P. Pentland. 3d structure from 2d motion. In *IEEE Signal Processing Magazine*, volume 16, 1999.

[15] R. Koch. Dynamic 3-D scene analysis through synthesis feedback control. *IEEE Pattern Analysis and Machine Intelligence*, 15(6):556–568, June 1993.

[16] H. Li, P. Roivainen, and R. Forchheimer. 3-D motion estimation in model-based facial image coding. *IEEE Pattern Analysis and Machine Intelligence*, 15(6):545–555, June 1993.

[17] R. McKendall and M. Mintz. Data fusion techniques using robust statistics. In M.A. Abidi and R.C. Gonzalez, editors, *Data Fusion in Robotics and Machine Intelligence*, 1992.

[18] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Pattern Analysis and Machine Intelligence*, 15(6):580–591, June 1993.

[19] Y. Moses, D. Reynard, and A. Blake. Robust real time tracking and classificiation of facial expressions. In *Proceedings ICCV '95*, pages 296–301, 1995.

[20] S. Negahdaripour. Revised definition of optical flow: Integration of radiometric and geometric cues for dynamic scene analysis. *IEEE Pattern Analysis and Machine Intelligence*, 20(9):961–979, September 1998.

[21] S. Negahdaripour and B. Horn. Direct passive navigation. *IEEE Pattern Analysis and Machine Intelligence*, 9(1):168–176, January 1987.

[22] A. Netravali and J. Salz. Algorithms for estimation of three-dimensional motion. *AT&T Technical Journal*, 64:335–346, 1985.

[23] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *IEEE Pattern Analysis and Machine Intelligence*, 13(7):715–729, 1991.

[24] J. Rehg, M. Loughlin, and K. Waters. Vision for a smart kiosk. In *Proceedings CVPR '97*, 1997.

[25] D. Reynard, A. Wildenberg, A. Blake, and J. Marchant. Learning dynamics of complex motions from image sequences. In *Proceedings ECCV '96*, pages I:357–368, 1996.

[26] G. Strang. *Linear algebra and its applications*. Harcourt, Brace, Jovanovich, 1988.

[27] H. Tao, H.H. Chen, W. Wu, and T.S. Huang. Compression of mpeg-4 facial animation parameters for transmission of talking heads. *IEEE Trans. Circuit ans Systems for Video Technology*, 9(2):264, March 1999.

[28] D. Terzopoulos and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Pattern Analysis and Machine Intelligence*, 15(6):569–579, 1993.

[29] D. Terzopoulos, A. Witkin, and M. Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123, 1988.